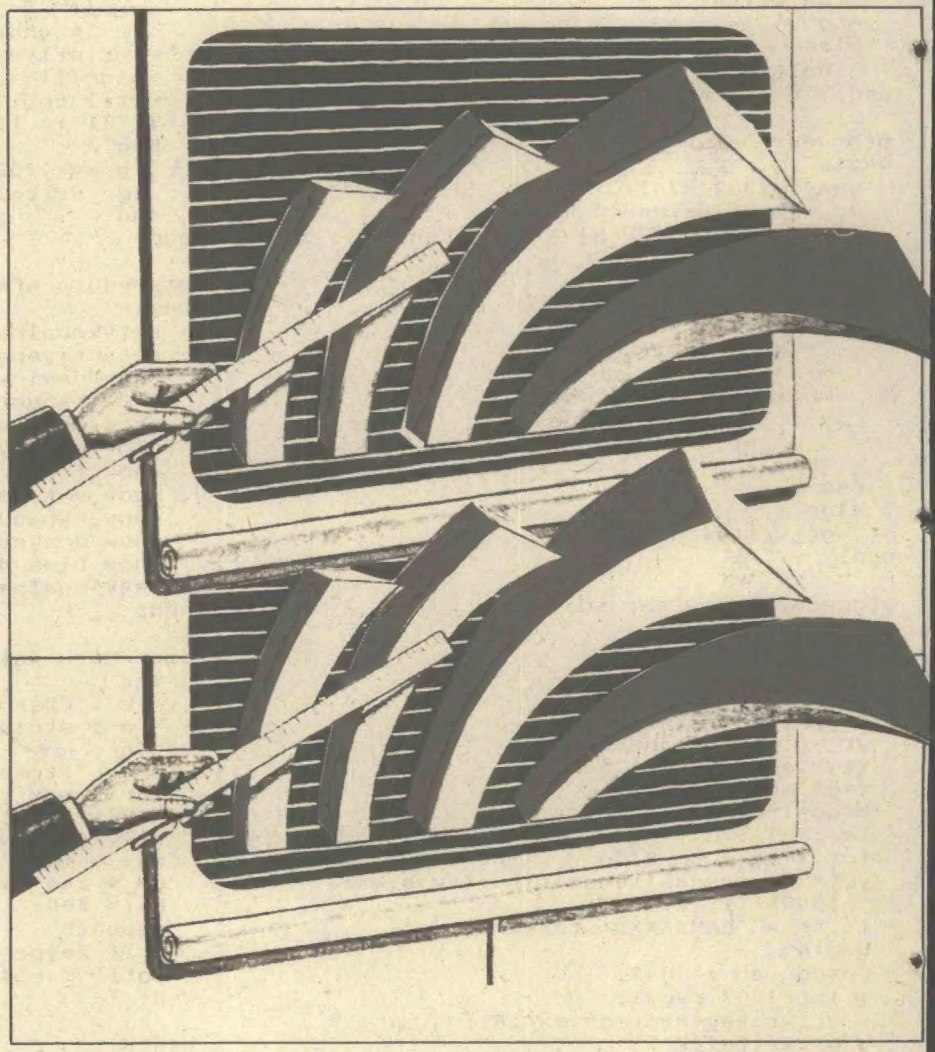
อาจหาญ สัตยารักษ์

SOFT MAIL

# ทางเลือกใหม่ในการจัดการฐานข้อมูลดีเบส (จบ)

***สนุกกับการสร้างโปรแกรมเขียนกราฟจากฐานข้อมูลในดีเบส ทั้งสวยและง่ายจนน่าทึ่ง***



ในการจัดการฐานข้อมูลดีเบสตอนสุดท้ายนี้จะเขียนถึงการนำข้อมูลที่มีอยู่มาเขียนกราฟ เพื่อจะได้รูปแบบที่แตกต่างจากในโปรแกรมดีเบส โดยแสดงเป็นตัวอย่างเพียง 2 แบบคือ กราฟวงกลมและกราฟแท่งเท่านั้น ผู้อ่านที่ประสงค์จะได้กราฟรูปแบบอื่นจะลองเขียนโปรแกรมเพิ่มเติมเองก็ย่อมได้

โปรแกรมสำหรับตอนนี้อยู่ในไฟล์ชื่อ dbase5.pas ผู้อ่านที่ไม่มีเวลาคีย์โปรแกรมก็ติดต่อผ่านทางซอฟต์เมล์ได้เช่นเคย

## แสดงข้อความกราฟิก

เพื่อให้เราสามารถแสดงข้อความกราฟิกบนจอภาพได้ตามความต้องการ ผู้เขียนได้นำโปรแกรมจากบทความเรื่อง *"การทำอักขระกราฟิกด้วย WORD PC"* ในวารสารไมโครฯฉบับที่ 52 หน้า 279--285 มาสร้างเป็นยูนิตชื่อว่า thaifont.tpu โดยเปลี่ยนชุดคำสั่งแสดงข้อความบนจอภาพจากของเดิมเป็น OutThaiXY แต่ยังคงใช้รหัสควบคุมแบบเดิมคือ

^PA-^PD ควบคุมรูปแบบการแสดงผล

^PK และ ^PL ควบคุมชนิดของฟอนต์ตัวหนังสือ เช่น ตัวปกติและตัวเอียงตามลำดับ

นอกจากนี้ยังได้เพิ่มรหัส ^PP เพื่อควบคุมสีของตัวอักขระบนจอภาพ หากตามหลังด้วยเลข 1-9 จะได้สีหมายเลข 1-9 แต่ถ้าปิดท้ายตัวเลขด้วย ^PP อีกครั้งหนึ่งจะได้สีหมายเลข 10-15 ทำให้การแสดงกราฟจากข้อมูลดีเบส สวยงามขึ้นเมื่อรันบนจอสี ส่วนการพิมพ์หน้าจอภาพผู้เขียนใช้โปรแกรมจากบทความเรื่อง *"พิมพ์กราฟิกด้วยปาสคาล 4.0"* ในฉบับที่ 44 หน้า 195-198 เป็นโปรแกรมดำเนินการ (ในชุดคำสั่งชื่อ hard copy)

## รูปแบบตัวแปรที่เปลี่ยนไป

เนื่องจากการใช้งานกราฟิกจำเป็นต้องใช้หน่วยความจำมาก ดังนั้นการกำหนดให้โปรแกรมที่อ่านได้จากแฟ้มเอกสารมีความยาวถึง 64 กิโลไบต์ อาจเกิดปัญหาหน่วยความจำฮีพไม่พอก็ได้ จึงต้องกำหนดตัวแปรเพื่อรองรับข้อมูลที่อ่านจากแฟ้มเอกสารใหม่ โดยลดความยาวลงเหลือเพียง 10000 ไบต์ก็เพียงพอต่อการใช้งาน ตัวแปรที่เพิ่มเติมมีดังนี้

- Count เป็นตัวแปรมิติ กำหนดไว้สำหรับนำข้อมูลตัวเลขที่ต้องการแสดงกราฟมาเก็บไว้

- LineGR ตัวแปรเก็บหมายเลขบรรทัดของแฟ้มเอกสารที่แสดงข้อมูลในการเขียนกราฟ

เราจำเป็นต้องเปลี่ยนแปลงชุดคำสั่ง ReadWord ใหม่ดังรูปที่ 1 ชุดคำสั่งหรือฟังก์ชันเดิมที่ต้องนำมาใช้งานคือ readchr, Read_DataBase และ DataInfield และฟังก์ชัน cut

```
PROGRAM ShowGraph_dBase;

uses  crt,dos,graph,ThaiFont;
type  HeapData=^WordData;
      WordData=record
                 WordF:array[0..65534] of char;
               end;
      HeapSTR=^String;
      Wordheap=^WordH;
      WordH=record
              wordd:array[0..10000] of char;
            end;
var
    ch,Funckey:char;  f:file;
    NoR,CountOfRecord,SizeOfRecord,SizeOfHead,EditNo:word;
    Head:array[0..1024] of byte;
    NameOfField:array[1..30] of string[10];
    TypeOfField:array[1..30] of char;
    WidthOfField:array[1..30] of byte;
    FDec:array[1..30] of byte;
    CountOfField:byte;
    Dbasefile,DateStr,DbaseSTR:string;
    UseDbf:boolean;
    FieldSTR:array[1..30] of string;  Space:string[40];
    DbaseNum,NumField:real;
    DataDbase:HeapData;
    Xstart,LineNo,LineMax,MaxF:word;
    WordST:array[1..60] of HeapSTR;
    NoF:array[1..60] of byte;  ck:integer;
    Xpoint,Ypoint:array[1..10] of word;
    Count:array[1..10] of word;
    LineGr:byte;  WorddBase:WordHeap;

PROCEDURE ReadWord(Filename:string);
var size:word;
begin
  assign(f,Filename); reset(f,1);
  size:=filesize(f);  blockread(f,WorddBase^,size);
  close(f);           LineNo:=1; Size:=0;
  repeat
   WordST[LineNo]^:='';
   repeat
    if WorddBase^.WordD[size] in [#1..#9,#11,#12,#14..#25,#28..#255] then
          WordST[LineNo]^:=WordST[LineNo]^+WorddBase^.WordD[size];
      inc(Size);
   until WorddBase^.wordD[size] in [#10,#26];
     LineMax:=LineNo; Inc(LineNo);
  until WorddBase^.wordD[size]=#26;
end;
```

รูปที่ 1 การกำหนดยูนิต ตัวแปร และชุดคำสั่ง ReadWord ที่ปรับปรุงใหม่

## แสดงข้อความบนจอภาพ

เพื่อความสะดวกในการกำหนดรูปแบบที่แสดงบนกราฟ ผู้เขียนจะใช้วิธีกำหนดรูปแบบผ่านแฟ้มเอกสารเขียนขึ้นด้วยโปรแกรมพิมพ์เอกสารทั่ว ๆ ไป ดังนั้นจึงต้องมีชุดคำสั่งในการนำข้อความที่กำหนดนั้นมาแสดงบนจอภาพให้ตรงกับความต้องการ วิธีการง่ายที่สุดก็คือกำหนดข้อความในลักษณะดังนี้

| ค่าแกน x, ค่าแกน y, ข้อความที่ต้องการแสดง |
|---|

เมื่ออ่านข้อความนี้ลงบนในหน่วยความจำแล้ว เราเพียงแต่เขียนโปรแกรมตรวจค่าหา x, y และข้อความที่ต้องการแสดงก็สามารถใช้คำสั่งจากยูนิต thaifont ที่ชื่อ OutThaiXY ทำการแสดงผลบนจอภาพได้ รูปที่ 2 เป็นชุดคำสั่งดังกล่าวโดยมีรายละเอียดต่อไปนี้

- ตรวจบรรทัดที่ 2 ของแฟ้มเอกสารว่าจะให้แสดงข้อความบนจอภาพกี่บรรทัด เราจะได้บรรทัดเริ่มต้น (Start) และบรรทัดสุดท้าย (En)

- ตรวจหาอักขระ , ในข้อความ (STT) แล้วนำข้อความที่อยู่ข้างหน้า (ในชุดคำสั่งคือตัวแปร ST) มาหาค่า x และตรวจหาอักขระ , ซ้ำอีกครั้งเพื่อนำไปหาค่า y จะได้ส่วนที่เหลือเป็นข้อความที่ต้องการแสดง โดยจะทำตั้งแต่บรรทัดหมายเลข Start ถึงบรรทัดหมายเลข En

- ในส่วนของข้อความนั้นสามารถใช้รหัส ^PA-^PD หรือ ^PK, ^PL กำหนดรูปแบบตัวอักษร

- หาค่าของตัวแปร LineGR เพื่อนำไปหาข้อมูลมาแสดงเป็นกราฟตามต้องการ

## รูปแบบของแฟ้มเอกสาร

กราฟอันดับแรกที่จะแสดงคือกราฟวงกลม ซึ่งอาศัยการกำหนดรูปแบบที่แฟ้มเอกสารอีกที โดยแฟ้มเอกสารที่จะนำมาใช้งานร่วมกับโปรแกรมภาษาปาสคาลนั้นผู้เขียนได้กำหนดรูปแบบไว้ดังนี้

1. บรรทัดที่ 1 ใช้กำหนดชื่อแฟ้มข้อมูลดีเบสและต้องระบุชื่อสกุลด้วย เช่น dbase5.dbf

2. บรรทัดที่ 2 ใช้กำหนดบรรทัดเริ่มต้นและบรรทัดสุดท้ายของข้อความที่ต้องการนำไปแสดงบนจอภาพ โดยเริ่มต้นที่บรรทัดที่ 3 ส่วนบรรทัดสุดท้ายจะจบที่ใดก็ได้ให้กำหนดเอง ตัวอย่างเช่น 3,9 (กำหนดแสดงข้อความ 7 บรรทัด) หรือ 3,5 (กำหนดแสดงข้อความ 3 บรรทัด) เป็นต้น

3. ตั้งแต่บรรทัดที่ 3 จนถึงบรรทัดสุดท้ายให้กำหนดรูปแบบดังที่กล่าวมาแล้ว

| 30,80,ผลการสอบรายวิชาฟิสิกส์ ว 021 |
|---|

ทั้งนี้ค่าตัวเลขที่กำหนดจะต้องอยู่ในกรอบภาพที่ใช้งาน เช่น 720×348 บนจอโมโนโครม เป็นต้น

4. บรรทัดถัดจากบรรทัดสุดท้าย (ตัวแปร LineGR) จะต้องมีรูปแบบดังนี้

| ชื่อเขตข้อมูล, หมายเลขระเบียนเริ่มต้น, หมายเลขระเบียนสุดท้าย<br>เช่น total,3,12<br>sum,1,6 เป็นต้น |
|---|

ซึ่งการพิมพ์จะต้องไม่เว้นช่องว่างไว้ (ถ้าต้องการให้ตรวจการเว้นช่องว่างในการป้อนจะต้องลองเขียนโปรแกรมเพิ่มเติมในบางส่วนโดยอาศัยฟังก์ชัน cut ช่วย)

5. บรรทัดสุดท้าย (บรรทัดที่ LineGr+1) ให้กำหนดดังนี้

| ค่าจุดศูนย์กลางแกน x,<br>ค่าจุดศูนย์กลางแกน y,<br>รัศมีของวงกลม<br>เช่น 200,160,30,300,160,120 เป็นต้น |
|---|

ตัวอย่างของแฟ้มเอกสารตามรูปแบบที่กำหนดจะเป็นดังรูปที่ 3 ซึ่งใช้กับแฟ้มข้อมูลชื่อ dbase5.dbf และนำข้อมูลจากเขตข้อมูลชื่อ sum ของระเบียนข้อมูลที่ 1-6 มาเขียนกราฟ โดยอาศัยชุดคำสั่งจากรูปที่ 4 ดำเนินการสร้างกราฟและแสดงข้อความตามที่กำหนด

## แสดงกราฟวงกลม

ผู้เขียนจะอธิบายชุดคำสั่ง ShowGraph_I ในรูปที่ 4 ดังนี้

- กำหนดรูปแบบของการระบายพื้นและสีของรูปด้วยตัวแปร Patt, CColor

- กำหนดมุมเริ่มต้นที่ 0 องศา (AngleSt=0)

- ทำการคำนวณหาผลรวมของข้อมูลในเขตข้อมูลที่กำหนด (sum) ซึ่งตอนนี้ถูกกำหนดเป็นตัวแปรมิติชื่อ Count [] แล้ว

- เริ่มเขียนส่วนของวงกลม (ใช้คำสั่ง pieslice) เริ่มคำนวณหามุมสุดท้าย (AngleEn) ก่อนเพื่อให้ได้ส่วนของวงกลมที่ต้องการ (โดยการกำหนดทั้งหมดต้องเขียนได้ครบรอบพอดี) ตามการกำหนดรูปแบบของพื้นที่ที่จะระบาย แล้วจึงเขียนส่วนของวงกลม

```
PROCEDURE ShowText;
var k,p,Start,En:byte; x,y:word; STT,ST:string;
begin
     STT:=CUT(WordST[2]^)+',';
     p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,start,ck);
     STT:=Copy(STT,p+1,length(STT)-p);
     p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,En,ck);
 for k:=start to En do
  begin  STT:=CUT(WordST[k]^);
       p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,x,ck);
       STT:=Copy(STT,p+1,length(STT)-p);
       p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,y,ck);
       STT:=Copy(STT,p+1,length(STT)-p);
       OutThaiXY(x,y,STT);
  end;
  LineGr:=En+1;
end;
```

รูปที่ 2 ชุดคำสั่งนำข้อความที่กำหนดด้วยแฟ้มเอกสารมาแสดงบนจอภาพ

```
dbase5.dbf
3,5
90,40,การแสดงกราฟวงกลมจากแฟ้มข้อมูลดีเบส
250,310,โดย อาจหาญ สัตยารักษ์
180,335,222 ถนนพหลโยธิน อ.เมือง ลำปาง 52000
sum,1,6
300,160,120
```

รูปที่ 3 ข้อความที่อยู่ในแฟ้มเอกสารชื่อ dbase5-1 สำหรับการแสดงกราฟวงกลม

- เปลี่ยนมุมเริ่มต้นมาเป็นมุมสุดท้ายของส่วนวงกลมที่เขียนเสร็จแล้ว

- ทำการเขียนส่วนของวงกลมในส่วนที่เหลือจนได้วงกลม 1 วงด้วยรูปแบบพื้นและสีที่ต่างกัน (จอโมโนโครมจะได้สีเดียว)

ส่วนชุดคำสั่ง Graphl มีรายละเอียดที่ต้องอธิบายดังต่อไปนี้

- อ่านแฟ้มเอกสารชื่อ dbase5-1

- หาชื่อแฟ้มข้อมูลดีเบสจากบรรทัดที่ 1 ของแฟ้มเอกสารเพื่ออ่านข้อมูลจากแฟ้มข้อมูลเก็บไว้ในหน่วยความจำ

- แสดงข้อความที่กำหนดจากแฟ้มเอกสารลงบนจอภาพ

- คำนวณหาหมายเลขของเขตข้อมูลที่กำหนด ซึ่งจะได้เขตข้อมูลชื่อ sum หมายเลขเขตข้อมูลที่ 5 ตามด้วยการคำนวณหาหมายเลขระเบียนเริ่มต้น (Start) และหมายเลขระเบียนสุดท้าย (En)

- นำข้อมูลจากเขตข้อมูล sum ตั้งแต่ระเบียนข้อมูลที่ Start ถึง En มาใส่ในตัวแปรมิติ Count []

- นำแฟ้มเอกสารบรรทัดถัดมา (บรรทัดที่ LineGr+1) ทำการหาจุดศูนย์กลางของวงกลมและรัศมี

- นำข้อมูลทั้งหมดที่ได้ไปแสดงกราฟจากการเรียกใช้ชุดคำสั่ง Show-Graph_I

จะได้ผลบนจอภาพ ดังรูปที่ 5

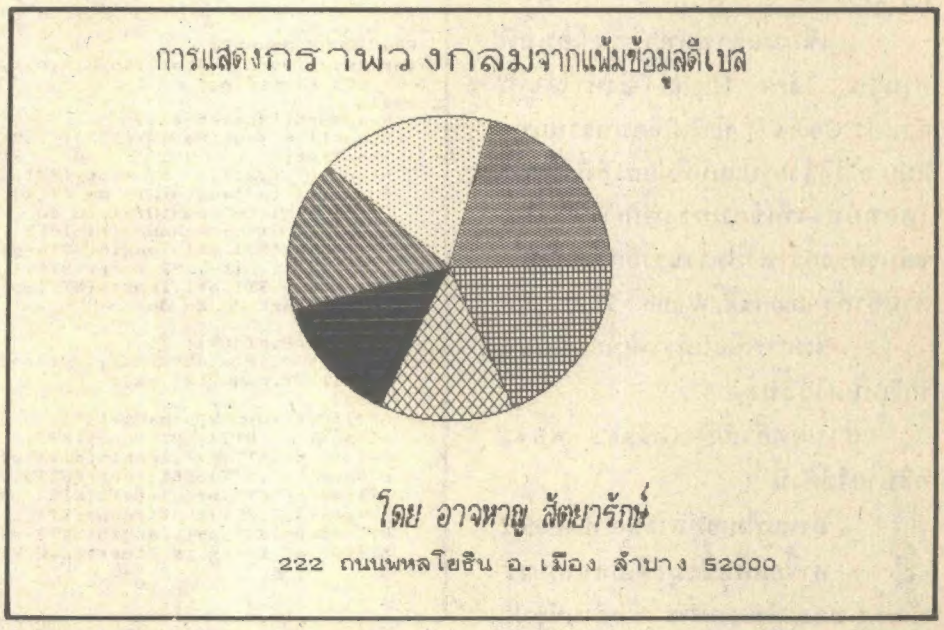
## แสดงกราฟแท่ง

กราฟแบบที่ 2 ผู้เขียนจะเสนอการแสดงข้อมูลแบบกราฟแท่งให้ดูเป็นตัวอย่างบ้าง โดยอาศัยแนวคิดเดิมคือ กำหนดรูปแบบผ่านแฟ้มเอกสารก่อนแล้วจึงเขียนโปรแกรมนำแฟ้มเอกสารนั้น ๆ มาแสดงผลบนจอภาพอีกที รูปที่ 6

```
PROCEDURE ShowGraph_I(Xpix,Ypix:word; NoSt,NoEnd:byte; R:word);
const Patt:array[1..6] of byte=(9,10,6,1,8,7);
      CColor:array[1..6] of byte=(5,4,15,2,14,3);
var k:byte;  sum,AngleSt,AngleEn:word;
begin
  setcolor(3);  Sum:=0;   AngleSt:=0;
  for k:=NoSt to NoEnd do  sum:=Sum+Count[k];
  for k:=NoSt to NoEnd do
   begin
    angleEn:=AngleSt+round(360/sum * Count[k]);
    if k=NoEnd then AngleEn:=360;
    setfillstyle(Patt[k-Nost+1],CColor[k-Nost+1]);
    pieslice(Xpix,Ypix,AngleSt,AngleEn,R);
    AngleSt:=AngleEn;
   end;
end;

PROCEDURE Graph1;
var xx,yy:word;  Start,En,R,p,k,No:byte;
    STT,ST:string;
begin
  ReadWord('dbase5-1');
  Dbasefile:=cut(WordST[1]^);   Read_DataBase;
  ShowText;                     STT:=Cut(WordST[LineGr]^);
  p:=pos(',',STT);  ST:=copy(STT,1,p-1);
  for k:=1 to length(ST) do ST[k]:=upcase(ST[k]);
    for k:=1 to CountOfField do
        if ST=copy(NameOfField[k],1,length(ST)) then No:=k;
  STT:=Copy(STT,p+1,length(STT)-p);
  p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,start,ck);
  STT:=Copy(STT,p+1,length(STT)-p);     Val(STT,En,ck);
 for k:=Start to En do
   begin
    DataInField(k);
    STT:=CUT(FieldSTR[no]); p:=pos('.',STT); STT:=Copy(STT,1,p-1);
    val(STT,count[k],ck);
   end;
  STT:=Cut(WordST[LineGr+1]^);
  p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,xx,ck);
  STT:=Copy(STT,p+1,length(STT)-p);
  p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,yy,ck);
  STT:=Copy(STT,p+1,length(STT)-p);     Val(STT,R,ck);
  ShowGraph_I(xx,yy,Start,En,R);
end;
```

รูปที่ 4 ชุดคำสั่งที่อ่านรูปแบบจากแฟ้มเอกสารและนำมาแสดงกราฟวงกลม




รูปที่ 5 ผลการแสดงกราฟวงกลมบนจอภาพ

คือตัวอย่างของแฟ้มเอกสาร ซึ่งถูกเก็บในชื่อ dbase5-2 โดยที่รูปแบบกำหนดยังคงเหมือนกับกราฟวงกลม แต่ต่างกันตรงบรรทัดสุดท้ายเท่านั้นที่กำหนดรูปแบบเป็นดังนี้

| |
|---|
| ค่าแกน x, ค่าแกน y,<br>ความสูง (สูงสุด) ของกราฟแท่ง,<br>ความกว้างของกราฟแท่ง<br>เช่น 50,300,150,60 เป็นต้น |

โดยมีความหมายว่า

50,300 เป็นจุดตัดของแกนนอนและแกนตั้งของกราฟที่จะแสดง

150 คือระยะสูงสุดของกราฟ (เมื่อเทียบค่ากับ 100)

60 คือความกว้างของแท่งกราฟ

ส่วนคำสั่งในการทำงานจะเป็นดังรูปที่ 7 ซึ่งผู้เขียนจะอธิบายรายละเอียดของชุดคำสั่ง ShowGraph_II ดังนี้

– สร้างแกนตั้งและแกนนอนแบบ 3 มิติด้วยคำสั่ง Bar3D

– เขียนเส้นบนแกนตั้งและให้ตัวเลขกำกับแกนตั้ง (ตามคำสั่งในช่วง for kk)

– เขียนกราฟแท่งตามข้อมูลที่คำนวณ ได้ค่า HighS ซึ่งหาได้จากตัวแปร Count [] นำไปเขียนกราฟแท่งแบบ 3 มิติด้วยรูปแบบพื้นและสีที่กำหนด โดยทุกครั้งที่เขียนกราฟเสร็จจะต้องขยับจุดลงกราฟไปทางขวาอีกระยะหนึ่งตามคำสั่ง Inc (xx, Width+20);

จบการเขียนกราฟเมื่อระเบียนที่กำหนดไว้สิ้นสุด

ส่วนชุดคำสั่ง Graph2 พอจะอธิบายได้ดังนี้

– อ่านแฟ้มเอกสารชื่อ dbase5-2

– หาชื่อแฟ้มข้อมูลดีเบสจากบรรทัดที่ 1 ของแฟ้มเอกสาร เพื่ออ่านข้อมูลจากแฟ้มข้อมูลเก็บไว้ในหน่วยความจำ

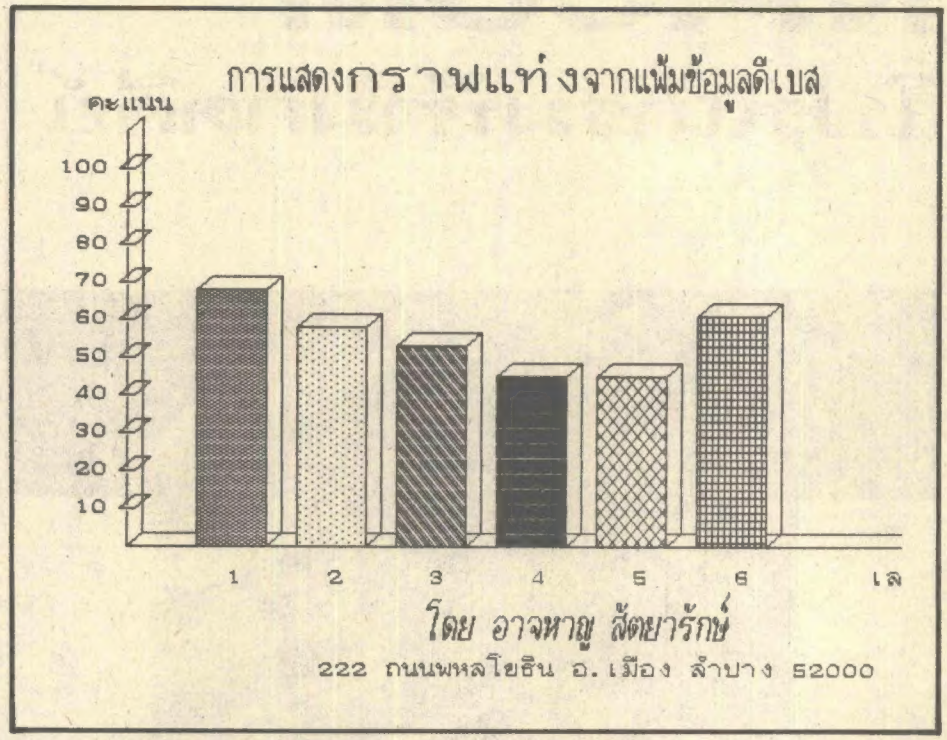
```
dbase5.dbf
3,7
120,30,การแสดงกราฟแท่งจากแฟ้มข้อมูลดีเบส
250,320,โดย อาจหาญ สัตยารักษ์
180,339,222 ถนนพหลโยธิน อ.เมือง ลำปาง 52000
30,40,คะแนน
120,290,1     2     3     4     5     6          เลขที่
sum,1,6
56,270,200,45
```

รูปที่ 6 ข้อความที่อยู่ในแฟ้มเอกสารชื่อ dbase5-2 สำหรับการแสดงกราฟแท่ง

```
PROCEDURE ShowGraph_II(Xpix,Ypix:word; NoSt,NoEnd:byte; High,Width:word)
const Patt:array[1..6] of byte=(9,10,6,1,8,7);
      CColor:array[1..6] of byte=(5,4,15,2,14,3);
var k,kk:byte;  XX,Y,yy,HighS:word;
    STT:string;
begin
    setcolor(3);
    bar3D(Xpix,Ypix-High-20,Xpix,Ypix,10,clipon);
    Bar3D(Xpix,Ypix,600,Ypix,10,clipon);
    XX:=Width; y:=High div 10; yy:=y;
  for kk:=1 to 10 do
    begin
       Bar3D(Xpix-5,Ypix-y,Xpix+5,Ypix-y,10,clipon);
       str(kk*10:3,STT); OutThaiXY(Xpix-45,Ypix-y+3,STT);
       Inc(y,yy);
    end;
  for k:=NoST to NoEnd do
   begin
    HighS:=Ypix-round(High/100 * Count[k]);
    setfillstyle(Patt[k-NoST+1],CColor[k-NoST+1]);
    Bar3D(Xpix+XX,Ypix,Xpix+xx+Width,Highs,10,clipon);
    inc(xx,width+20);
   end;
end;

PROCEDURE Graph2;
var xx,yy:word;  Start,En,H,W,p,k,No:byte;
    STT,ST:string;
begin
  ReadWord('dbase5-2');
  Dbasefile:=cut(WordST[1]^);    Read_DataBase;
  ShowText;                      STT:=Cut(WordST[LineGr]^);
  p:=pos(',',STT);  ST:=copy(STT,1,p-1);
  for k:=1 to length(ST) do ST[k]:=upcase(ST[k]);
    for k:=1 to CountOfField do
        if ST=copy(NameOfField[k],1,length(ST)) then No:=k;
  STT:=Copy(STT,p+1,length(STT)-p);
  p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,start,ck);
  STT:=Copy(STT,p+1,length(STT)-p);     Val(STT,En,ck);
 for k:=Start to En do
   begin
    DataInField(k);
    STT:=CUT(FieldSTR[no]); p:=pos('.',STT); STT:=Copy(STT,1,p-1);
    val(STT,count[k],ck);
   end;
  STT:=Cut(WordST[LineGr+1]^);
  p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,xx,ck);
  STT:=Copy(STT,p+1,length(STT)-p);
  p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,yy,ck);
  STT:=Copy(STT,p+1,length(STT)-p);
  p:=pos(',',STT); ST:=copy(STT,1,p-1); Val(ST,H,ck);
  STT:=Copy(STT,p+1,length(STT)-p);     Val(STT,W,ck);
  ShowGraph_II(xx,yy,Start,En,H,W);
end;
```

รูปที่ 7 ชุดคำสั่งที่อ่านรูปแบบจากแฟ้มเอกสารนำมาแสดงเป็นกราฟแท่ง




รูปที่ 8 ภาพที่ได้จากการแสดงกราฟแท่งของชุดคำสั่งในรูปที่ 7

```
PROCEDURE ClsGR(x1,y1,x2,y2:word);
begin
   setfillstyle(1,0); Bar(x1,y1,x2,y2);
   setfillstyle(1,0);
end;

{=============== ShowGraph dBase =================}

begin
   opengraph(1);
   Fillchar(Space,41,#32); Space[0]:=#40;
   for NoR:=1 to 30 do                          begin
      Mark(WordST[NoR]); new(WordST[NoR]);  end;
      Mark(DatadBase);   new(DatadBase);
      Mark(WorddBase);   new(WorddBase);
      Graph1;
      readchr;  if Funckey=#59 then hardcopy;
      ClsGR(0,0,GetMaxX,GetMaxY);
      Graph2;
      readchr;  if Funckey=#59 then hardcopy;
   release(DatadBase); release(worddBase);
   for NoR:=1 to 30 do release(WordST[NoR]);
  closegraph;
end.
```

รูปที่ 9 ส่วนของโปรแกรมหลักในการแสดงกราฟวงกลมและกราฟแท่ง

– แสดงข้อความที่กำหนดจากแฟ้มเอกสารลงบนจอภาพ

– คำนวณหาหมายเลขของเขตข้อมูลที่กำหนดซึ่งจะได้เขตข้อมูลชื่อ sum หมายเลขเขตข้อมูลที่ 6 ตามด้วยการคำนวณหาหมายเลขระเบียนเริ่มต้น (Start) และหมายเลขระเบียนสุดท้าย (En)

– นำข้อมูลจากเขตข้อมูล sum ตั้งแต่ระเบียนข้อมูลที่ Start ถึง En มาใส่ในตัวแปรมิติ Count []

– นำแฟ้มเอกสารบรรทัดต่อไป (บรรทัดที่ LineGr+1) มาหาจุดตัดของแกนนอนและแกนตั้ง (xx,yy) ความสูงของแท่งกราฟ (H) และความกว้างของแท่งกราฟ (W)

– นำข้อมูลทั้งหมดที่ได้ไปแสดงกราฟจากการเรียกใช้ชุดคำสั่ง Show-Graph_II

เราจะได้ผลบนจอภาพดังรูปที่ 8

## โปรแกรมหลัก

ในรูปที่ 9 มีชุดคำสั่งสำหรับลบจอภาพชื่อ ClsGR เพื่อสั่งลบจอภาพทั้งหมดหากต้องการเขียนกราฟรูปต่อไป นอกนั้นเป็นการเรียกใช้ชุดคำสั่งให้แสดงกราฟพร้อมกับรอกดแป้น โดยถ้าประสงค์จะพิมพ์ภาพหน้าจอลงกระดาษก็ให้กดแป้น F1 เราจะได้ภาพตามต้องการ

ส่วนในซอฟต์เมล์นั้นผู้เขียนได้นำทุกชุดคำสั่งที่ใช้งานมารวมไว้ด้วยกันแล้ว จึงสะดวกในการศึกษามาก สำหรับการเขียนกราฟในรูปแบบอื่น ขอให้ผู้อ่านทดลองเขียนโปรแกรมขึ้นเองดีกว่า คิดว่าคงจะสามารถทำได้ไม่ยากนัก เพราะบทความนี้ได้นำร่องเป็นตัวอย่างมาพอสมควร จึงหวังว่าบทความชุดนี้คงจะช่วยให้ผู้อ่านใช้ประโยชน์จากแฟ้มข้อมูลดีเบสของตนได้มากขึ้น

*ผู้เขียนคงต้องบอกว่าลาที..แต่มิใช่ลาก่อน*